NEW YORK CITI BIKE

Cece Li, Jackie Tang, Lucas Balsinde, Zhizhuo Li

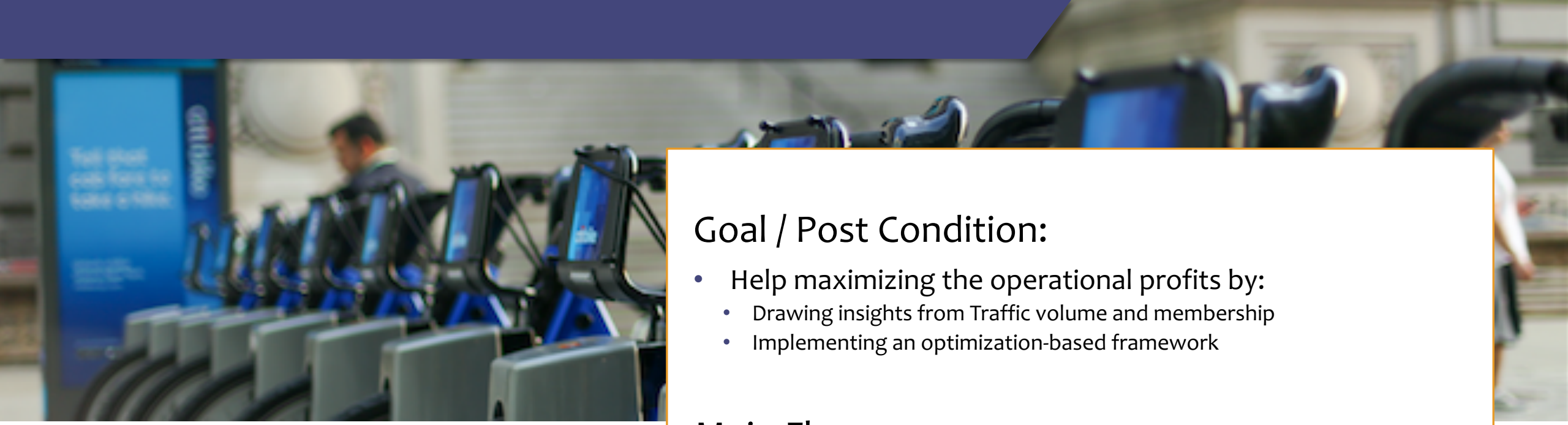
EXECUTIVE SUMMARY

- The rapid development of a bike-sharing company in New York City and its wealth of public datasets has drawn data professionals to address the user demand and resource allocation challenges. The available datasets open up the opportunity of deep geographical, time, and pricing analysis.

- This analysis aims to help Citi Bike maximize profits by implementing an optimization-based framework. we will identify patterns to determine whether their current resource allocation and pricing is profit maximizing, and how they can gain market share in this competitive industry.

AGENDA

- Executive Summary
- Business Use Case
- Data Sources / Tools
- Dimensional Model
- Data Preparation
- Design Considerations
- Data Loading
- SQL Queries
- Tableau Dashboard



BUSINESS USE CASE

Goal / Post Condition:

- Help maximizing the operational profits by:
  - Drawing insights from Traffic volume and membership
  - Implementing an optimization-based framework

Main Flows:

- Analyze the data
- Analyze the business model
- Consider the COVID risk as both a concern and an opportunity
- Rebalance organizational resources
- Adjust the pricing and resource allocation

DATA SOURCES

- Citi Bike System Data
- Bus Stop locations
- NYC Weather Data
- NYC Health: COVID-19 Data

DATA & TOOLS

Data cleaning and wrangling

- Clean, organize, and enhance
- Prevent redundancies
- Increase Efficiency and understandability

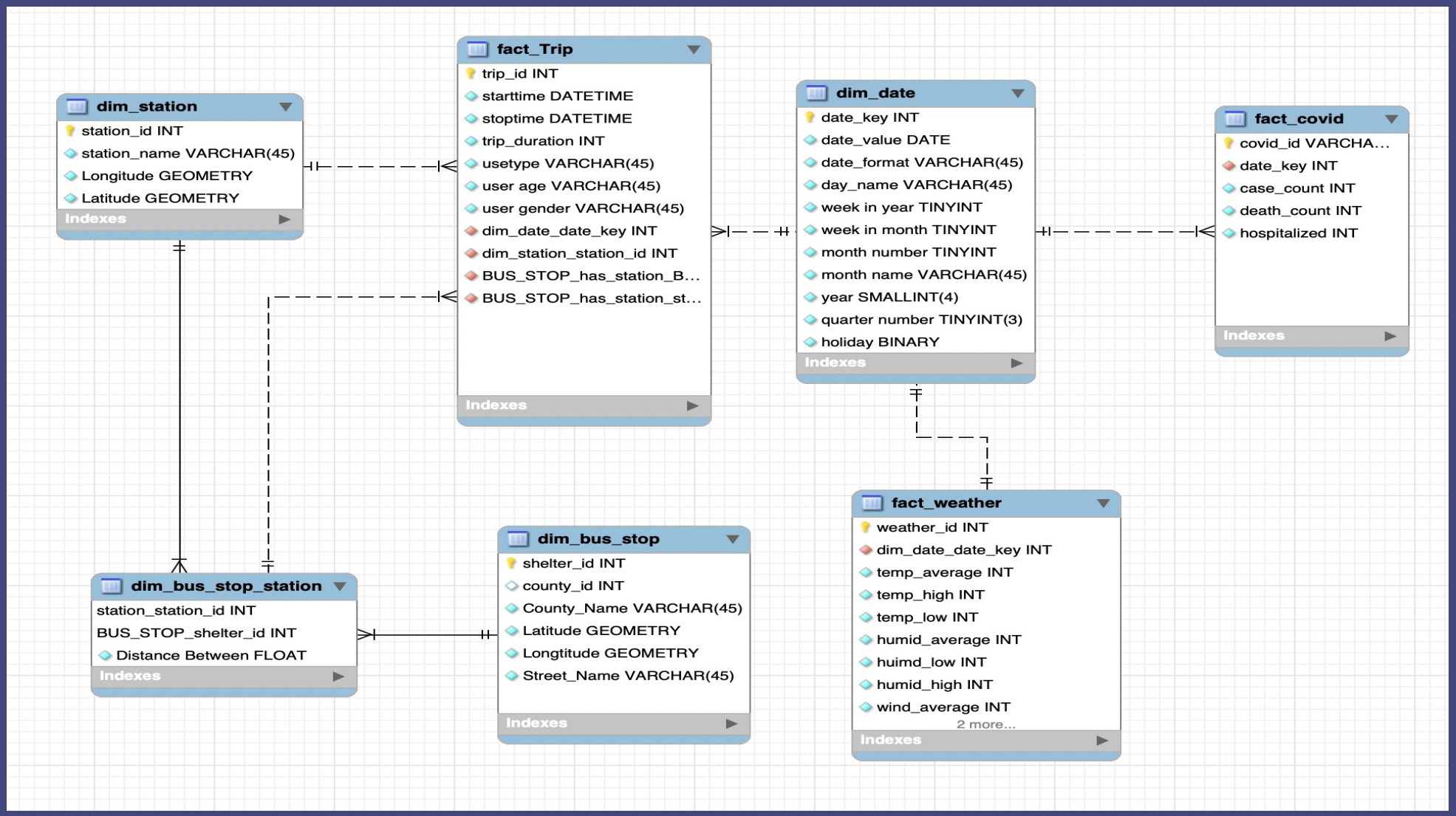
Table relationship building

- Merge and combine the datasets
- Create a comprehensive database

Data Analysis and visualization

- Analyze the business models
- Come up with better solution for resource allocation
- Visualize the results from data analysis
- Create user-friendly dashboard for different audiences

DIMENSIONAL MODEL



Data Preparation

- Data Sampling
  - Randomly select 0.1% of 40 million rows
  - 1.5 years

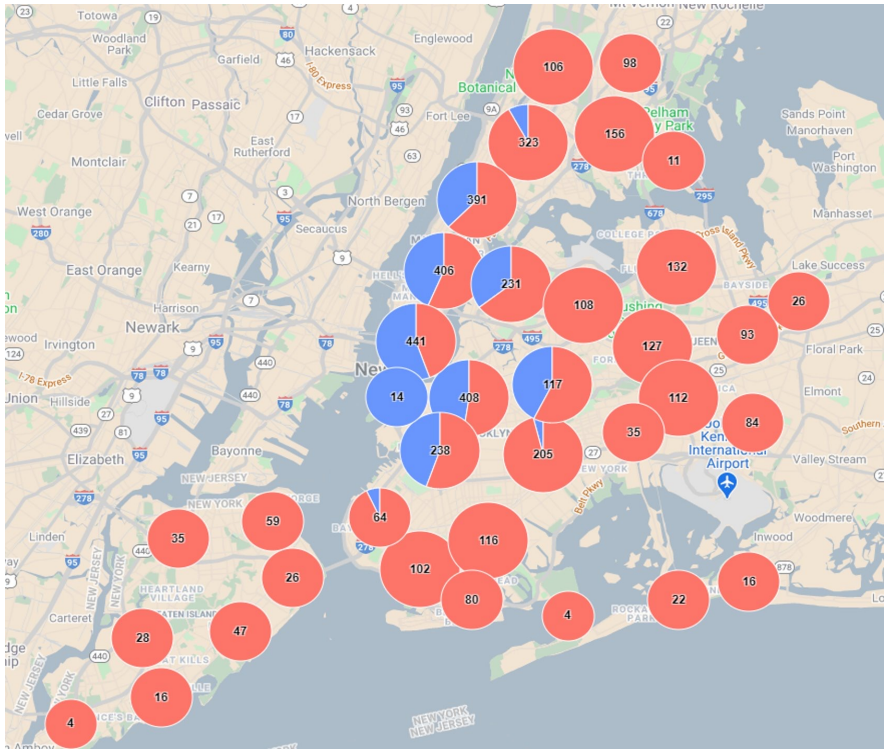
- Data Cleaning
  - Fill missing values
  - Remove the unrelated data columns
  - Split and format the columns

- Data Structure & Organization
  - Unified format to SQL analytics uses

```
import os
import pandas as pd

folder = 'C:/Users/Jackie/Desktop/Data/Citibikedata'
data_list = []
for data in os.listdir(folder):
    df = pd.read_csv(folder+'/'+data).sample(frac=0.01)
    data_list.append(df)
    print(f'finished {len(data_list)}')

print(f'finished or not: {len(data_list)==17}')
final_dataset = pd.concat(data_list)
```



Platform Considerations

- OLAP / Dimensional Model
- Google Cloud
- Multiple fact tables
  - fact_Trip
  - fact_covid
  - fact_weather
- Dimension tables
  - Dim_bus_stop / Dim_station / Dim_bus_stop_station
  - Dim_date / Dim_time
  - Dim_date connects to three fact tables - fact_trip, fact_weather, and fact_covid

Other Design Considerations

- The use of multiple fact tables:
  - Weather and covid data contain measures rather than categorical variables
  - Unique id of fact_weather and fact_covid

- Dimensional model vs ER model:
  - The data contain lots of historical data
  - Standard framework and easier for user to understand
  - Being able to accommodate unexpected new data elements and design changes.

Loading Data & Calculations

- Loading data
  - Mainly Import directly
  - Fix the data format
  - The order of importing based on dependency
  - Dimension tables vs. Fact table

- Dim bus_stop_station
  - The joint table between bus stop and Citi bike station
  - Insert shelter id (bus stop) and station id (citi_bike)
  - Set default values for distance as NULL initially
  - Use function to calculate the distance

```
INSERT INTO dim_bus_stop_station(has_station_id,has_shelter_id)
SELECT  dim_station.station_id,
        dim_bus_stop.shelter_id
FROM
        dim_station
CROSS JOIN
        dim_bus_stop;

UPDATE citi_bike.dim_bus_stop_station
SET  `Distance Between` = (
       SELECT ST_Distance_Sphere(
          point(bus.Longtitude, bus.Latitude),
          point(station.Longitude, station.Latitude))
       FROM
        dim_bus_stop AS bus,
        dim_station AS station
       WHERE
        bus.shelter_id = citi_bike.dim_bus_stop_station.has_shelter_id
            AND
        station.station_id = citi_bike.dim_bus_stop_station.has_station_id);
```

MySQL Queries

I. Factors affect trips of citi bike (weather and covid)

WEATHER & CITI BIKE

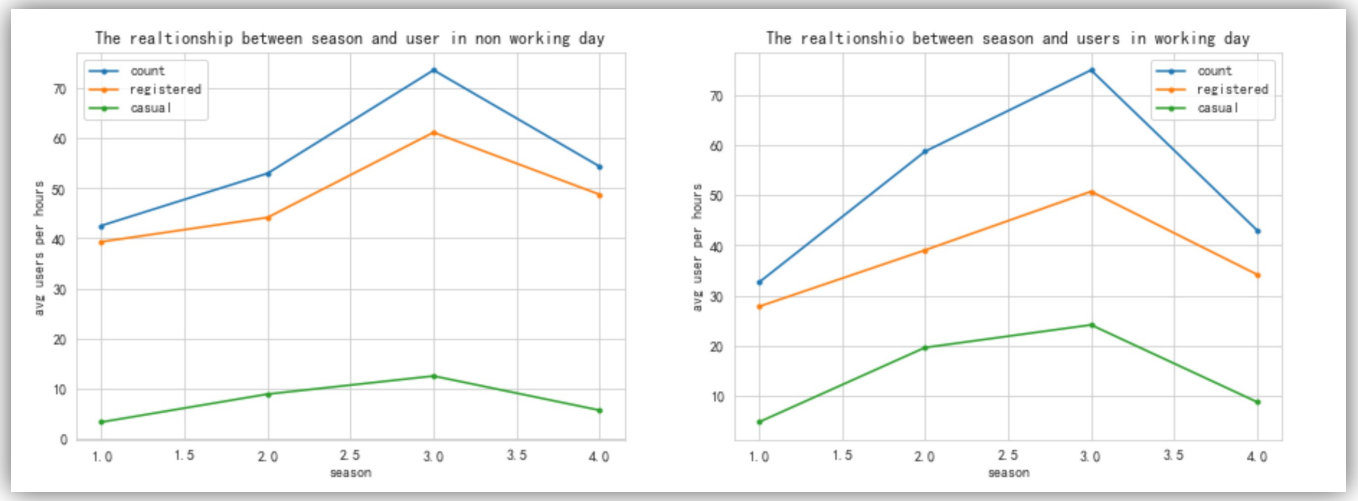
```sql
select
    trip_date,
    trip_cnt,
    temp_average,
    humid_average,
    wind_average
from
    (select
        count(distinct trip_id) as trip_cnt,
        trip_date
    from
        fact_Trip
    group by
        trip_date) trip_cnt
        inner join
    fact_weather wea on trip_cnt.trip_date = wea.dim_date_date_value
order by
    trip_date, trip_cnt desc;
```

Spring

trip_date	trip_cnt	temp_average	humid_average	wind_average
2019-04-03	62	52	26	17
2019-04-04	58	50	16	14
2019-04-05	40	40	26	10
2019-04-06	46	50	39	7
2019-04-07	75	56	40	8
2019-04-08	57	57	49	12
2019-04-09	64	46	42	9
2019-04-10	66	51	31	13
2019-04-11	68	45	21	7
2019-04-12	68	54	44	12
2019-04-13	74	64	56	9
2019-04-14	66	60	57	10
2019-04-15	39	55	46	19
2019-04-16	65	53	23	18
2019-04-17	70	53	31	8
2019-04-18	48	56	45	12
2019-04-19	64	63	57	14

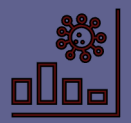
Summer

trip_date	trip_cnt	temp_average	humid_average	wind_average
2019-07-30	74	87	68	9
2019-07-31	65	79	67	7
2019-08-01	101	81	59	7
2019-08-02	91	78	62	9
2019-08-03	95	79	67	10
2019-08-04	60	79	64	6
2019-08-05	78	76	59	10
2019-08-06	80	75	68	8
2019-08-07	56	77	69	10
2019-08-08	71	77	64	8
2019-08-09	73	78	59	9
2019-08-10	90	76	51	12
2019-08-11	59	76	51	7
2019-08-12	61	80	56	8
2019-08-13	73	78	67	7
2019-08-14	87	76	68	9
2019-08-15	79	76	64	8
2019-08-16	77	75	66	8
2019-08-17	87	77	70	8
2019-08-18	73	80	71	8
2019-08-19	76	83	69	8



```sql
select
    weekofyear(trip.trip_date) as week_no,
    count(distinct trip_id) as trip_cnt,
    sum(case_count) as total_case,
    sum(death_count) as total_death,
    sum(hospitalized) as total_hospitalized
from
      fact_trip trip
        inner join
      fact_covid covid on trip.trip_date = covid.date_value
group by week_no
order by week_no;
```

COVID & CITI BIKE



week_no	trip_cnt	total_case	total_death	total_hospitalized
9	77	33	0	0
10	337	2050	0	1749
11	374	133179	348	34941
12	177	517787	4907	97861
13	104	475641	18874	146997
14	130	646776	62322	211877
15	145	702765	82083	209633
16	172	547354	74394	146239
17	167	417413	50272	84069
18	241	393497	50064	68459
19	225	240277	29616	41992
20	358	295466	29729	43938
21	328	272492	18992	32524
22	430	237115	19200	32086
23	332	143804	10968	21566
24	512	168201	13350	27107
25	438	137948	8604	21309
26	460	143293	10141	20336
27	462	145598	8573	18323
28	426	151881	5579	17560
29	509	183437	5551	20390
30	491	134081	4543	16765
31	495	125757	2875	17232
32	503	120224	2651	15508
33	498	123949	2314	15820
34	553	126589	2420	17265
35	552	138363	3043	14905

COMPARE TRIP COUNTS IN 2019 VS 2020

```
select
    dim_date.`month name`,
    `month number`,
    weekofyear(trip.trip_date) as week_no,
    count(IF(year(trip.trip_date)=2019,1,null)) AS Count_2019,
    count(IF(year(trip.trip_date)=2020,1,null)) AS Count_2020
from
    fact_Trip trip
    left join
    dim_date on dim_date.date_value = trip.trip_date
where `month number` is not null
    and dim_date.holiday is false
group by dim_date.`month number`,dim_date.`month name`, week_no
order by dim_date.`month number`;
```

month name	month number	week_no	Count_2019	Count_2020
January	1	1	174	119
January	1	2	275	319
January	1	3	206	274
January	1	4	178	277
January	1	5	112	224
February	2	5	72	60
February	2	6	270	305
February	2	7	228	215
February	2	8	197	269
February	2	9	150	262
March	3	9	83	44
March	3	10	238	337
March	3	11	334	374
March	3	12	291	177
March	3	13	382	104
March	3	14	0	28
April	4	14	406	102
April	4	15	463	145
April	4	16	366	172
April	4	17	422	167
April	4	18	107	95
May	5	18	264	146
May	5	19	418	168
May	5	20	438	358
May	5	21	491	328
May	5	22	252	357
June	6	22	149	0
June	6	23	600	332
June	6	24	449	512
June	6	25	450	368
June	6	26	476	460
June	6	27	0	138
July	7	27	395	195
July	7	28	530	380
July	7	29	481	509
July	7	30	491	491
July	7	31	223	353
August	8	31	347	142
August	8	32	507	503
August	8	33	537	498
August	8	34	510	553
August	8	35	443	552
August	8	36	0	78

II. Trip pattern and user analysis

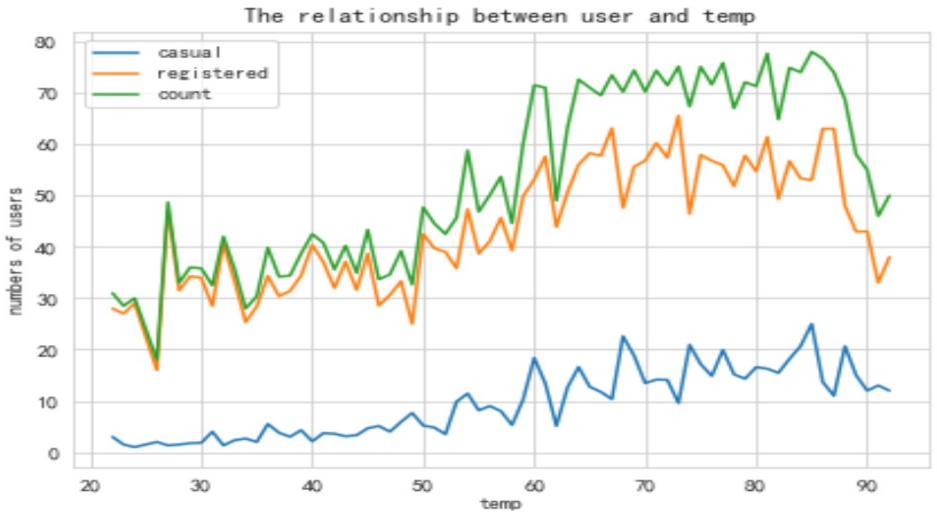
```sql
select
    `month name`,
    `month number`,
    `quarter number`,
    year,
    count(distinct trip_id) as trip_cnt,
    round(avg(trip_mins), 3) as avg_trip_mins,
    round(avg(trip_distance), 3) as avg_distance,
    round(avg(`user age`), 3) as avg_usr_age,
    round(avg(`user gender`), 3) as avg_usr_gender,
    count(if(usetype = 'Subscriber', 1,null)) as subscriber_trip,
    count(if(usetype = 'Customer', 1,null)) as customer_trip
from
    (select
        trip_id,
        trip_duration/60 as trip_mins,
        `user age`,
        `user gender`,
        trip_distance,
        trip_date ,
        usetype
    from
        fact_Trip) trip_usr
            inner join
        dates on trip_usr.trip_date = dates.date_value
group by `month name`, `month number`, `quarter number`, year
order by year,`month number`;
```

month name	month number	quarter number	year	trip_cnt	avg_trip_mins	avg_distance	avg_usr_age	avg_usr_gender	subscriber_trip	customer_trip
January	1	1	2019	967	11.621	0.924	41.099	1.195	918	49
February	2	1	2019	944	11.620	0.991	41.075	1.168	894	50
March	3	1	2019	1328	11.681	1.019	40.98	1.155	1225	103
April	4	2	2019	1764	14.987	1.104	40.516	1.153	1530	234
May	5	2	2019	1924	15.952	1.151	40.258	1.173	1653	271
June	6	2	2019	2124	20.955	1.157	39.142	1.164	1742	382
July	7	3	2019	2177	15.876	1.123	39.516	1.151	1814	363
August	8	3	2019	2344	22.889	1.12	39.238	1.161	1905	439
September	9	3	2019	2444	18.247	1.117	38.98	1.171	2033	411
October	10	4	2019	2093	14.947	1.126	38.84	1.161	1771	322
November	11	4	2019	1478	12.582	1.015	39.815	1.194	1320	158
December	12	4	2019	955	12.759	0.963	40.546	1.178	881	74
January	1	1	2020	1241	13.253	1.011	40.274	1.184	1144	97
February	2	1	2020	1146	12.380	1.024	39.706	1.188	1052	94
March	3	1	2020	1064	49.195	1.176	40.108	1.166	907	157
April	4	2	2020	681	31.009	1.281	39.069	1.203	506	175
May	5	2	2020	1487	28.769	1.342	38.978	1.159	975	512
June	6	2	2020	1880	22.911	1.387	38.486	1.182	1313	567
July	7	3	2020	2103	21.876	1.324	37.963	1.165	1545	558
August	8	3	2020	2326	19.184	1.327	38.11	1.146	1664	662

BASIC INFORMATION ABOUT CITI BIKE TRIPS & USER PROFILE IN EACH MONTH

CUSTOMER VS SUBSCRIBER

```
select
    dim_date.`month name`,
    dayofweek(trip.trip_date) as week_no,
    count(IF(usetype='Subscriber',1,null)) AS Count_Subscriber,
    count(IF(usetype='Customer',1,null)) AS Count_Customer
from
    fact_Trip trip
    left join
dim_date on dim_date.date_value = trip.trip_date
where `month number` is not null
    and  dim_date.holiday is false
group by dim_date.`month number`,dim_date.`month name`, week_no
order by dim_date.`month number`;
```



month name	week_no	Count_Subscriber	Count_Customer
January	1	231	31
January	2	223	18
January	3	313	11
January	4	361	22
January	5	372	19
January	6	336	19
January	7	180	22
February	1	209	22
February	2	259	18
February	3	317	15
February	4	297	17
February	5	308	9
February	6	272	12
February	7	230	43
March	1	268	57
March	2	301	27
March	3	352	36
March	4	327	25
March	5	310	16
March	6	321	46
March	7	253	53
April	1	239	97
April	2	295	34
April	3	386	53
April	4	345	53
April	5	292	49
April	6	233	28
April	7	246	95
May	1	254	166
May	2	232	38
May	3	356	71
May	4	419	77
May	5	418	61
May	6	464	103
May	7	364	197
June	1	386	182
June	2	447	121
June	3	457	106
June	4	451	90

USERS INFO

```sql
select
    case when `user gender`=1 then 'male'
         when `user gender`=2 then 'female'
         when `user gender`=0 then 'unknown' end gender,
    usetype,
    count(case when `user age` <= 18 then 1 else null end) as teen_cnt,
    count(case when `user age` between 19 and 65 then 1 else null end) as adult_cnt,
    count(case when `user age` >=66 then 1 else null end) as senior_cnt
from
    fact_trip
group by `user gender`, usetype
order by gender;
```

gender	usetype	teen_cnt	adult_cnt	senior_cnt
female	Subscriber	12	6812	168
female	Customer	15	1271	4
male	Subscriber	33	18799	522
male	Customer	31	1958	8
unknown	Subscriber	0	438	8
unknown	Customer	0	2391	0

III. Comparison of trips in weekdays and weekends

WEEKDAYS VS WEEKENDS

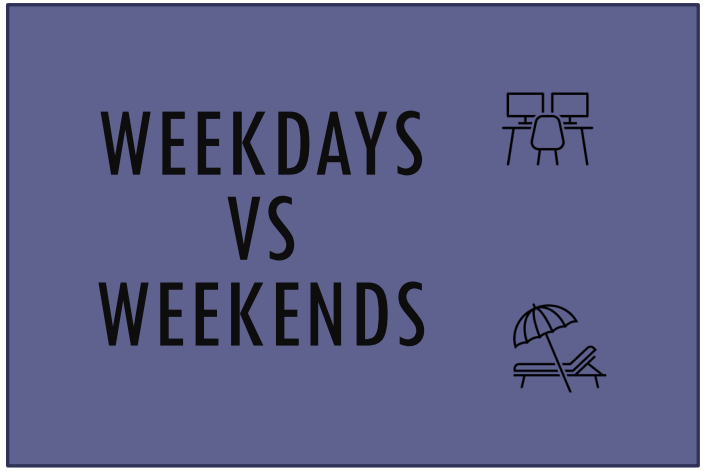
```sql
select
    case when dayofweek(trip_date) in (2,3,4,5,6) then 'weekday'
         when dayofweek(trip_date) in (1,7) then 'weekend' end day_in_week,
    dayofweek(trip_date) as weekday_n,
    count(distinct trip_id) as trip_cnt,
    round(avg(trip_distance),3) as distance,
    round(avg(trip_duration/60),3) as trip_mins
from
    fact_trip
group by day_in_week, weekday_n
order by weekday_n;
```

Trip Count

day_in_week	weekday_n	trip_cnt	distance	trip_mins
weekend	1	4246	1.203	32.210
weekday	2	4441	1.145	15.197
weekday	3	4794	1.135	15.253
weekday	4	4878	1.141	14.284
weekday	5	4680	1.134	15.194
weekday	6	4663	1.102	16.342
weekend	7	4768	1.208	25.259

COMPARE DESTINATION OF TRIPS ON WEEKDAYS AND WEEKENDS

```sql
select
    count(trip.trip_id) as trip_cnt,
    case when dayofweek(trip.trip_date) in (2,3,4,5,6) then 'weekdays'
         when dayofweek(trip.trip_date) in (1,7) then 'weekends' end day_in_week,
    bus.county_Name
from
    fact_Trip trip
    left join
(select
    has_station_id,
    has_shelter_id,
    `Distance Between`
from
    dim_bus_stop_station
where `Distance Between` in (select min(`Distance Between`)
                             from dim_bus_stop_station
                            group by has_station_id)) as near
    on near.has_station_id = trip.end_station_id
    left join
dim_bus_stop bus on bus.shelter_id = near.has_shelter_id
group by day_in_week, bus.county_Name
order by trip_cnt desc;
```



Destination

trip_cnt	day_in_week	county_Name
18565	weekdays	Manhattan
6512	weekends	Manhattan
4193	weekdays	Brooklyn
2109	weekends	Brooklyn
697	weekdays	Queens
385	weekends	Queens
27	weekdays	Bronx
15	weekends	Bronx

TOP 10 POPULAR START STATIONS ON WEEKDAYS AND WEEKENDS

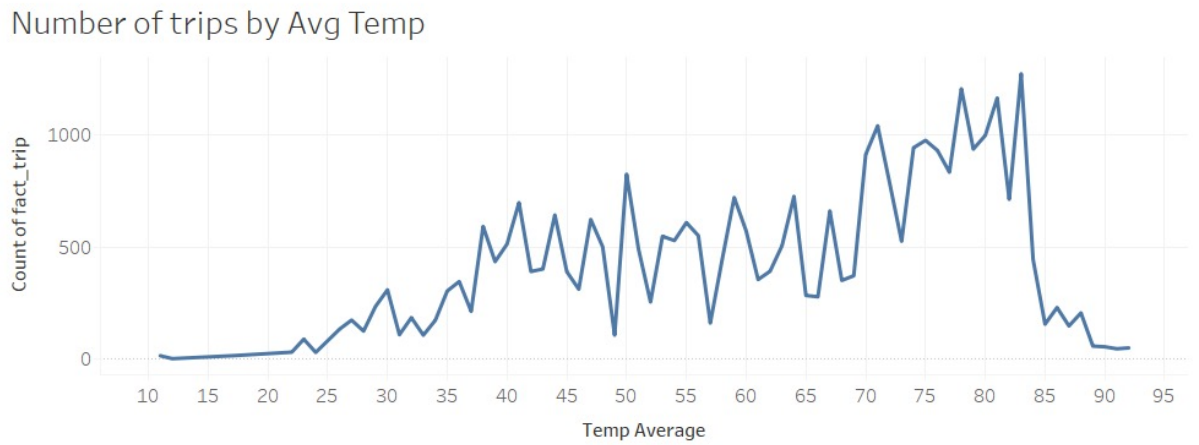
```
(select
    station.station_name,
    count(trip.trip_id) as trip_cnt,
    case when dayofweek(trip.trip_date) in (2,3,4,5,6) then 'weekdays'
         when dayofweek(trip.trip_date) in (1,7) then 'weekends' end day_in_week
from
    fact_Trip trip
    left join
dim_station station on trip.start_station_id = station.station_id
group by station.station_name, day_in_week
having day_in_week = 'weekdays'
order by trip_cnt desc
limit 10)
union
(select
    station.station_name,
    count(trip.trip_id) as trip_cnt,
    case when dayofweek(trip.trip_date) in (2,3,4,5,6) then 'weekdays'
        when dayofweek(trip.trip_date) in (1,7) then 'weekends' end day_in_week
from
    fact_Trip trip
    left join
dim_station station on trip.start_station_id = station.station_id
group by station.station_name, day_in_week
having day_in_week = 'weekends'
order by trip_cnt desc
limit 10) ;
```

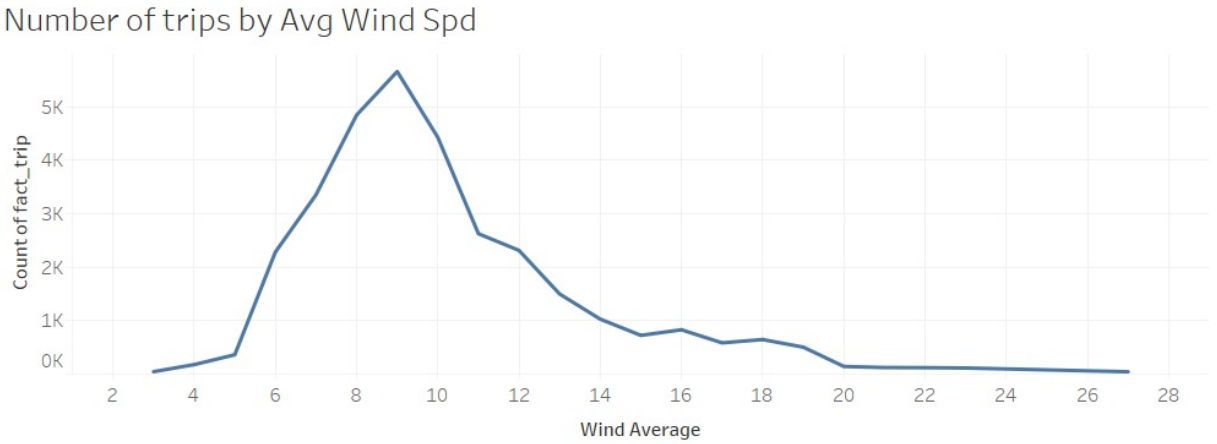
station_name	trip_cnt	day_in_week
Pershing Square North	186	weekdays
Broadway & E 22 St	136	weekdays
Broadway & E 14 St	123	weekdays
8 Ave & W 31 St	119	weekdays
E 17 St & Broadway	117	weekdays
Christopher St & Greenwich St	116	weekdays
1 Ave & E 68 St	115	weekdays
Broadway & W 25 St	114	weekdays
8 Ave & W 33 St	113	weekdays
West St & Chambers St	112	weekdays
12 Ave & W 40 St	66	weekends
Broadway & W 60 St	59	weekends
W 21 St & 6 Ave	57	weekends
West St & Chambers St	57	weekends
Christopher St & Greenwich St	50	weekends
W 20 St & 11 Ave	46	weekends
5 Ave & E 73 St	46	weekends
Pier 40 - Hudson River Park	44	weekends
E 17 St & Broadway	44	weekends
Cleveland Pl & Spring St	44	weekends

Tableau Dashboard

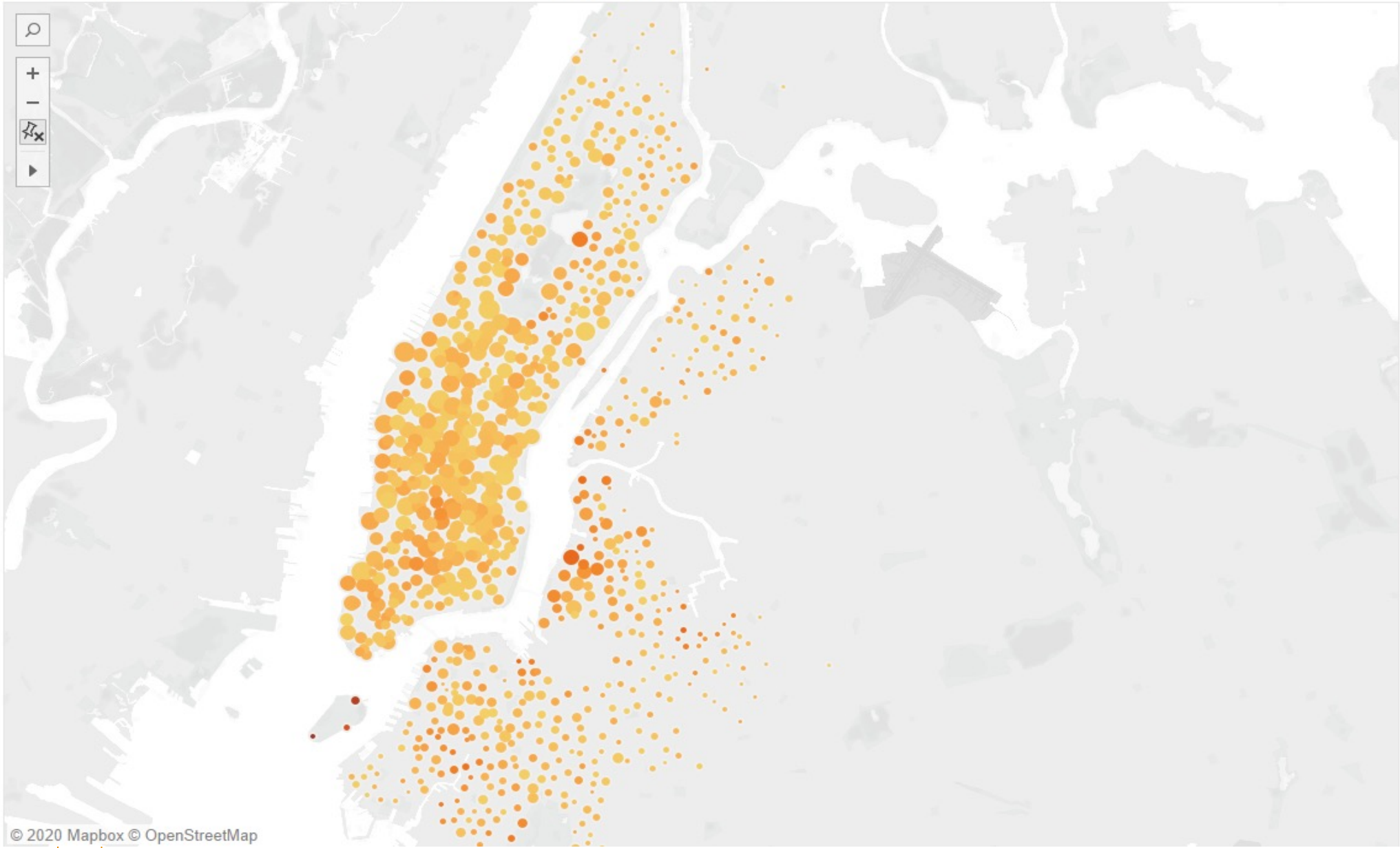
RIDERSHIP PATTERNS PRE-COVID

We want to find the conditions in which bikes are most highly demanded

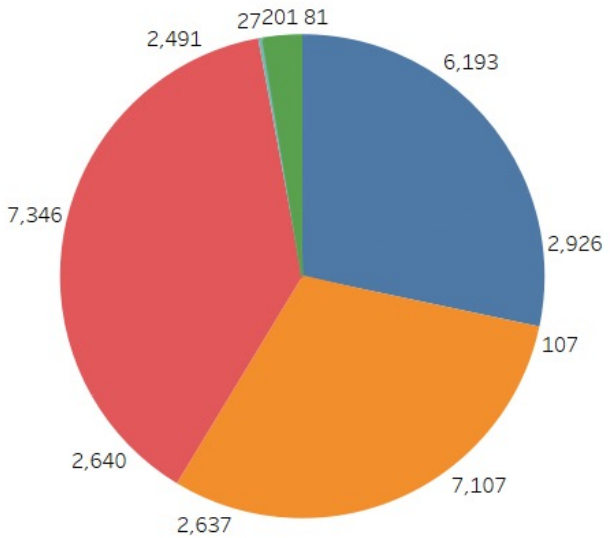




Ridership by station and distance to nearest bus stop



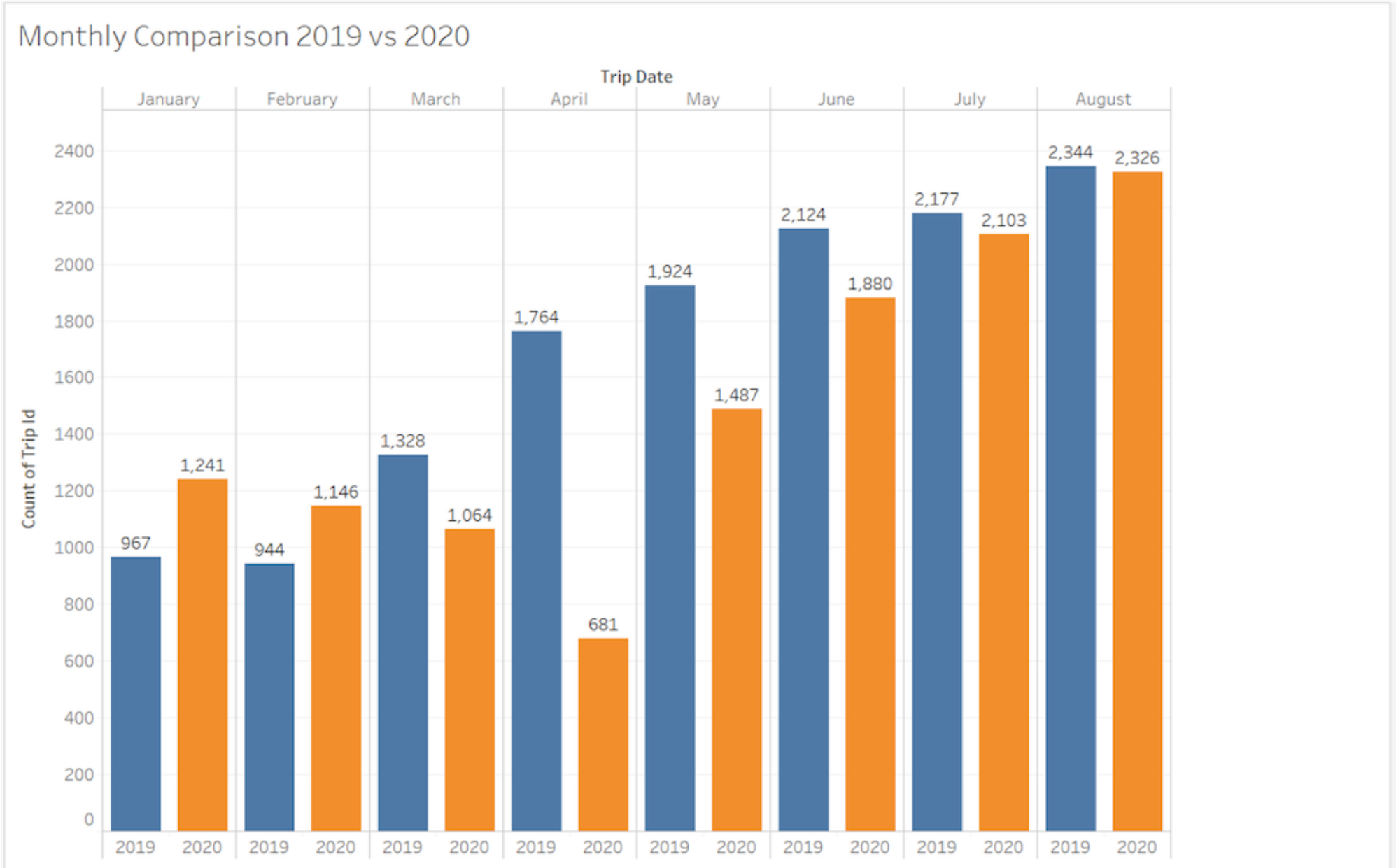
Proportion of trips by Age Group and Gender

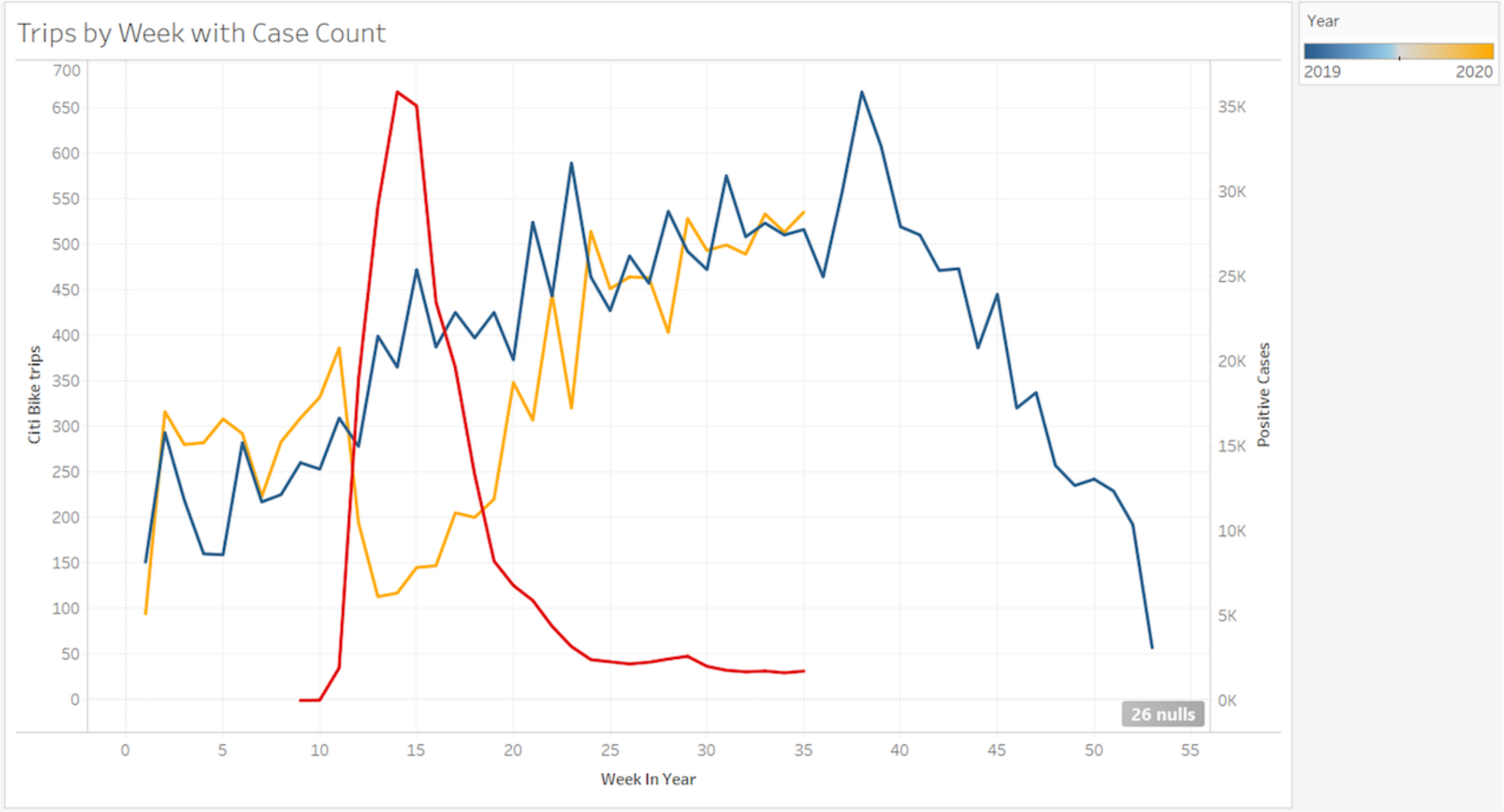


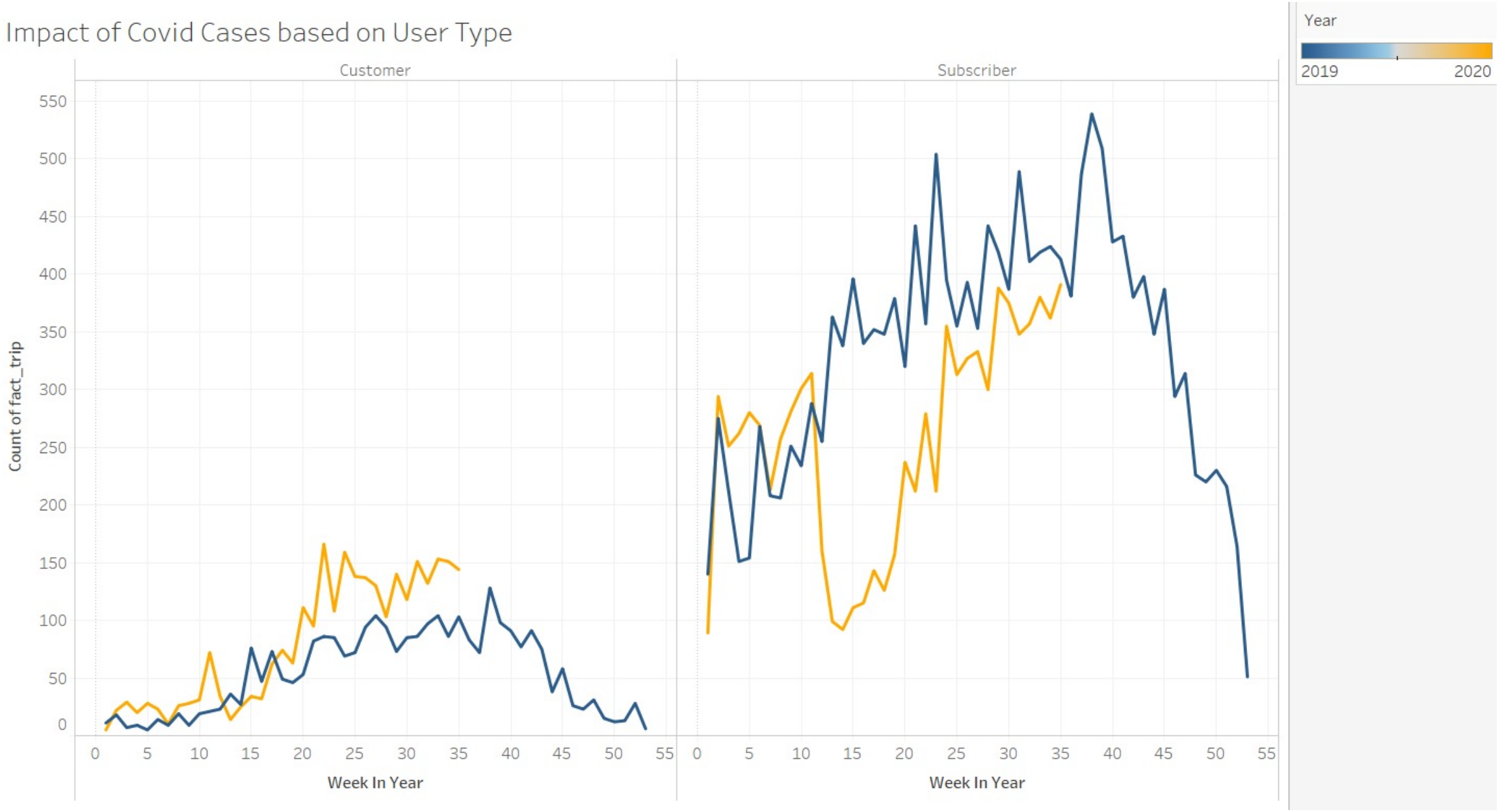
User Gender
- (All)
- 0
- 1
- 2

Age Group
- 19-30
- 30-45
- 45-64
- <18
- Seniors

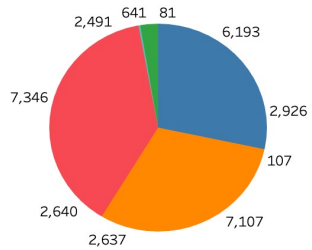
Most popular stations


Monthly Comparison 2019 vs 2020
Trip Date
January
February
March
April
May
June
July
August
Count of Trip Id
2400
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
967
1,241
944
1,146
1,328
1,064
1,764
681
1,924
1,487
2,124
1,880
2,177
2,103
2,344
2,326
2019
2020
YEAR(Trip Date)
2019
2020

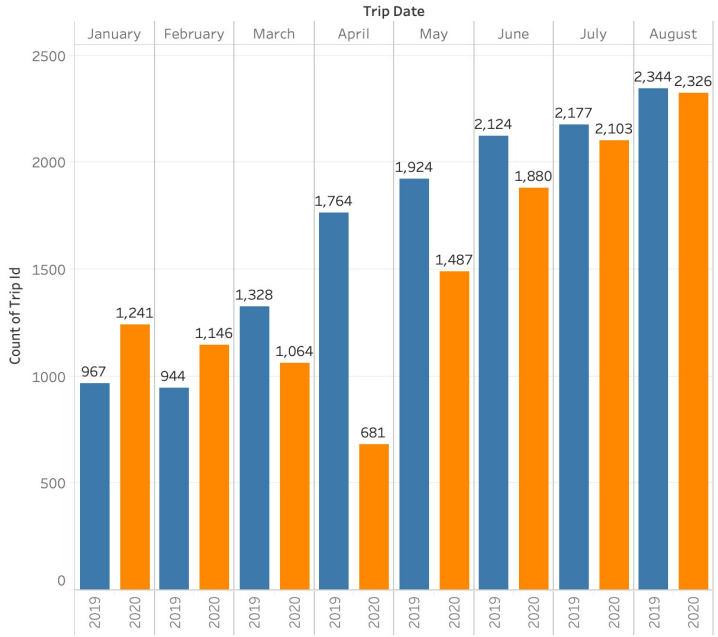

Trips by Week with Case Count
Year
2019
2020
Citi Bike trips
Positive Cases
Week In Year
26 nulls
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
0K
5K
10K
15K
20K
25K
30K
35K
0
5
10
15
20
25
30
35
40
45
50
55


Impact of Covid Cases based on User Type
Customer
Subscriber
Count of fact_trip
550
500
450
400
350
300
250
200
150
100
50
0
0 5 10 15 20 25 30 35 40 45 50 55
Week In Year
Year
2019
2020

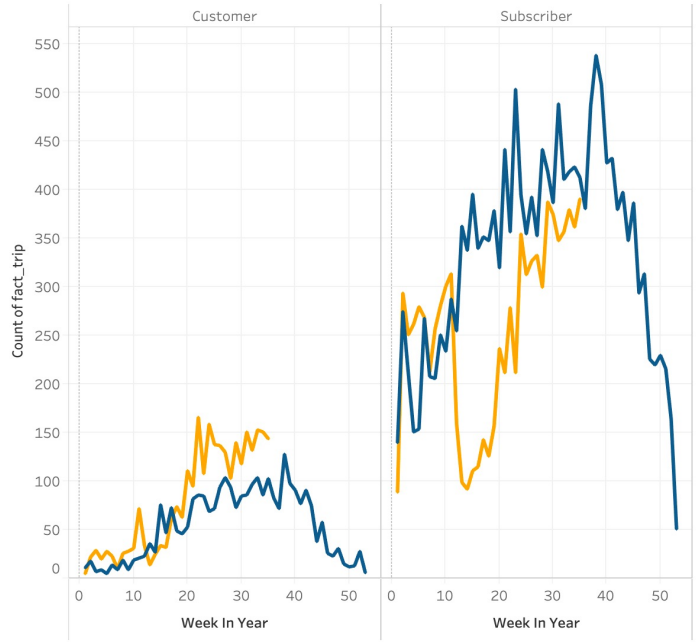
Proportion of trips by Age Group and Gender



Monthly Comparison 2019 vs 2020



Impact of Covid Cases based on User Type



Count by station
0
50
100
150
200
226

Mininum Distance
0.0029
0.8530

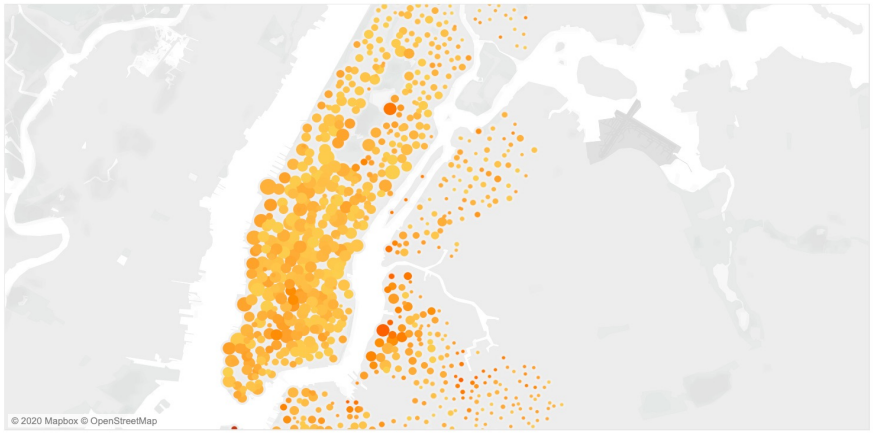
Year of Trip Date
2019
2020

User Gender
(All)
0
1
2

Age Group
19-30
30-45
45-64
<18
Seniors

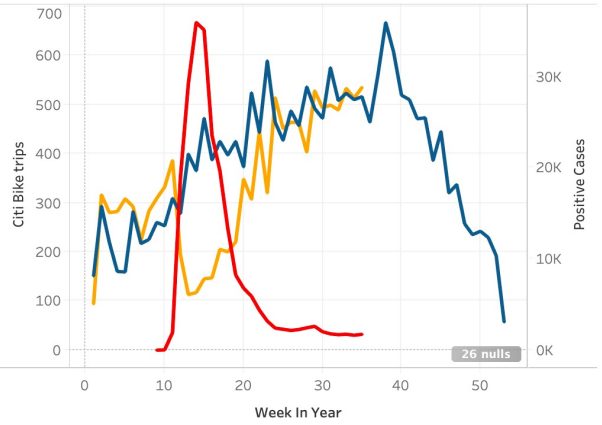
Year
2019
2020

Ridership by station and distance to nearest bus stop



Most popular stations

Trips by Week with Case Count

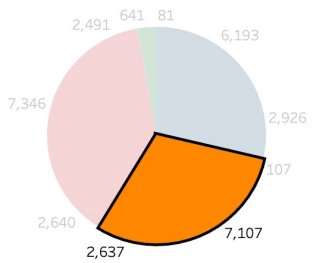


Filter

Year
2019
2020

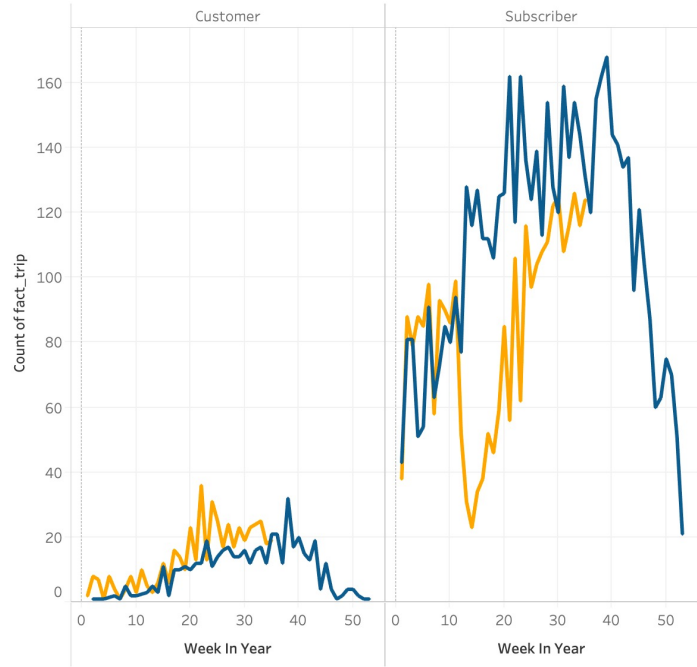
Count of fact_trip
100
226

Proportion of trips by Age Group and Gender



Monthly Comparison 2019 vs 2020

Impact of Covid Cases based on User Type



Count by station
1
50
100
150
200
226

Mininum Distance
0.0029
0.7585

Year of Trip Date
2019
2020

User Gender
(All)
0
1
2

Age Group
19-30
30-45
45-64
<18
Seniors

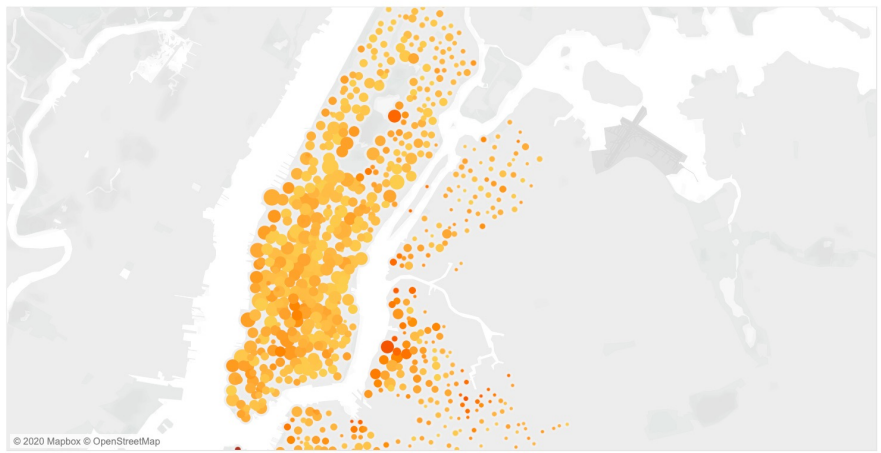
Year
2019
2020

Filter

Year
2019
2020

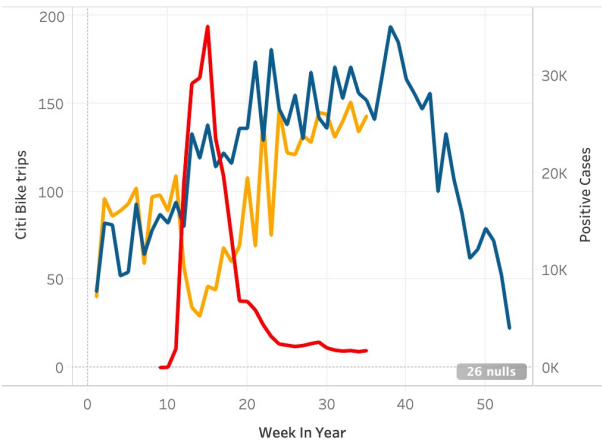
Count of fact_trip
Null

Ridership by station and distance to nearest bus stop

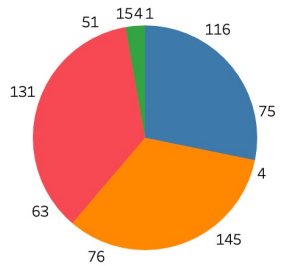


Most popular stations

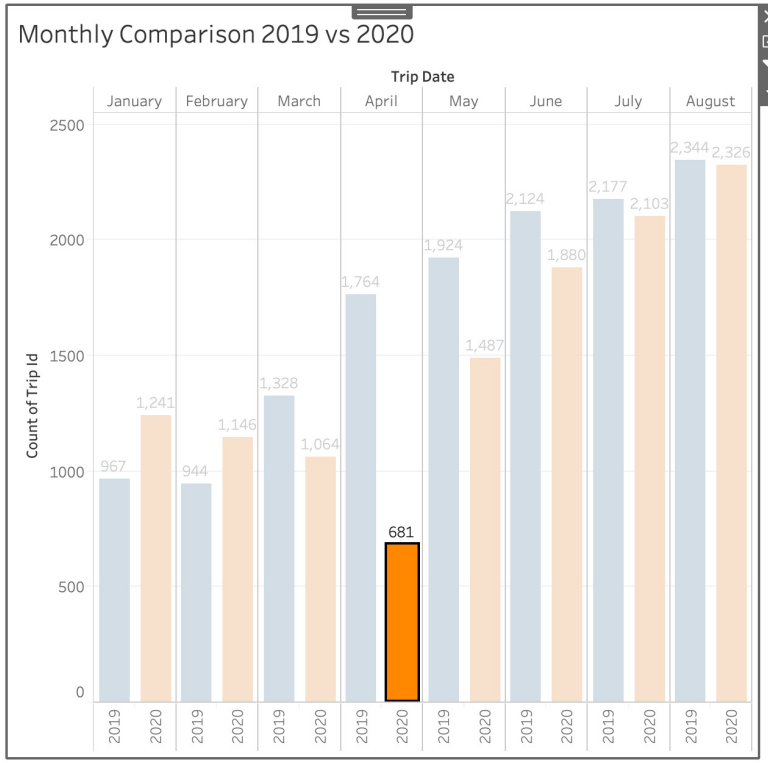
Trips by Week with Case Count



Proportion of trips by Age Group and Gender



Monthly Comparison 2019 vs 2020



Impact of Covid Cases based on User Type

Count by station
1
50
100
150
200
226

Mininum Distance
0.0029
0.4675

Year of Trip Date
2019
2020

User Gender
(All)
0
1
2

Age Group
19-30
30-45
45-64
Seniors

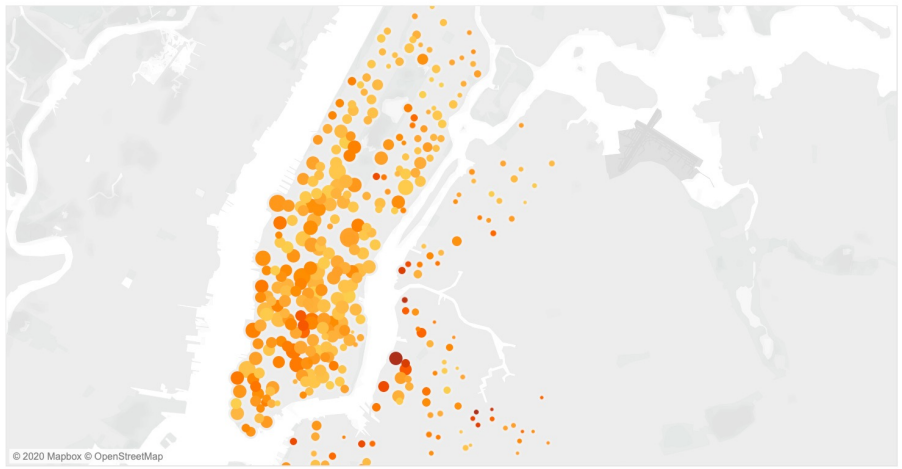
Year
2020

Filter

Year
2020

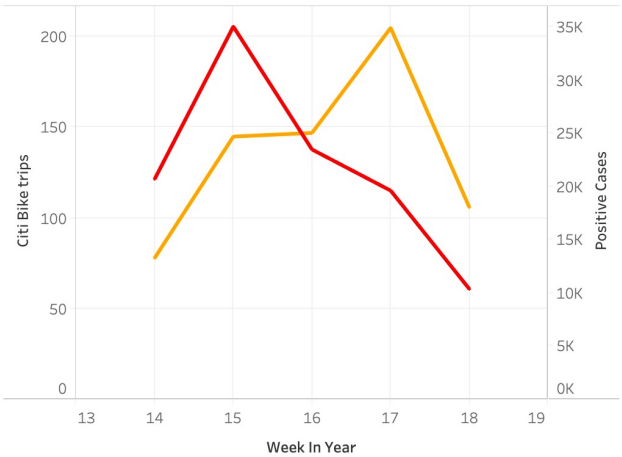
Count of fact_trip
Null

Ridership by station and distance to nearest bus stop

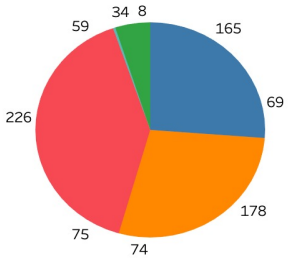


Most popular stations

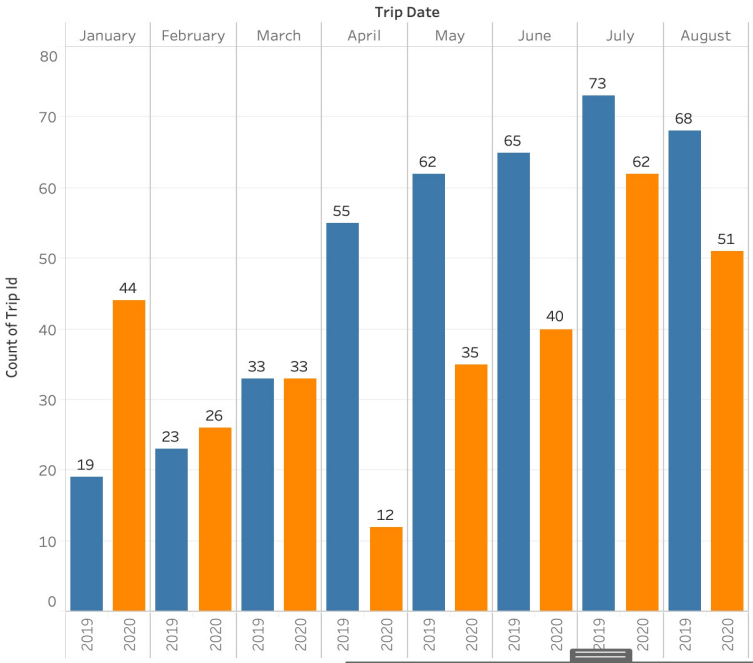
Trips by Week with Case Count



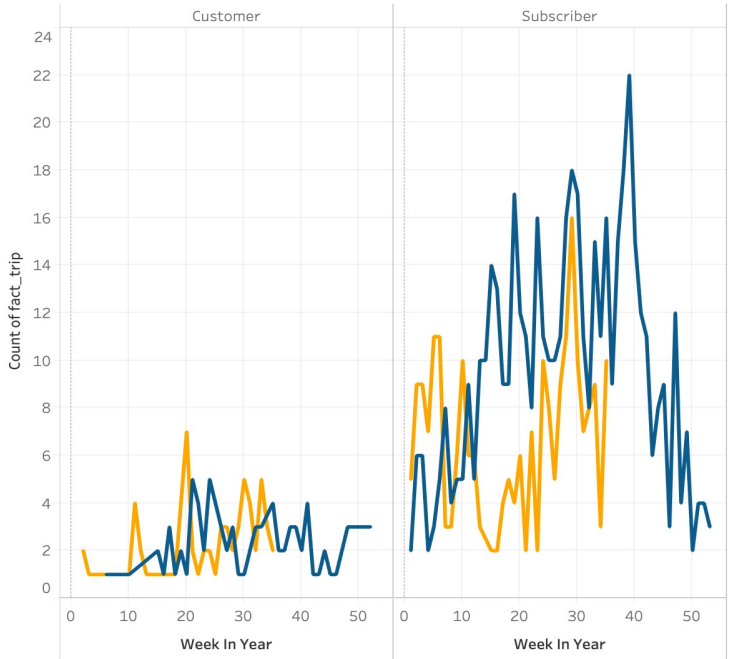
Proportion of trips by Age Group and Gender



Monthly Comparison 2019 vs 2020



Impact of Covid Cases based on User Type



Count by station
161
180
200
226

Mininum Distance
0.01056 0.10365

Year of Trip Date
2019
2020

User Gender
(All)
0
1
2

Age Group
19-30
30-45
45-64
<18
Seniors

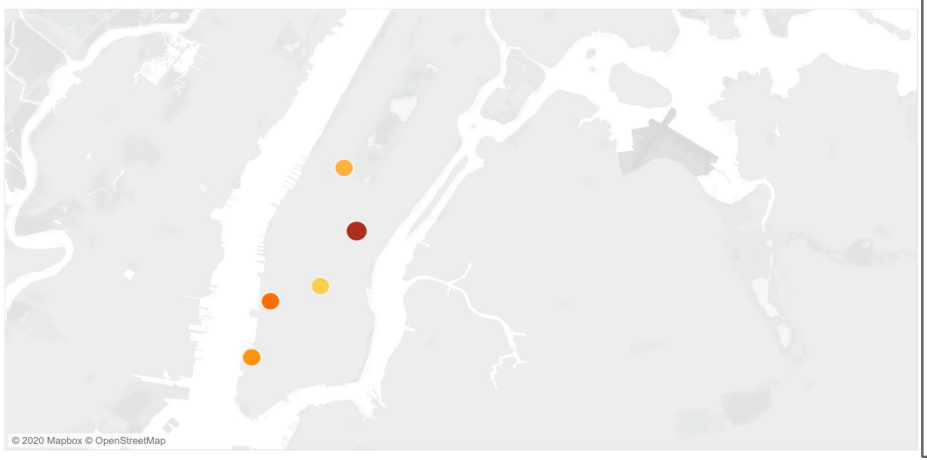
Year
2019 2020

Filter

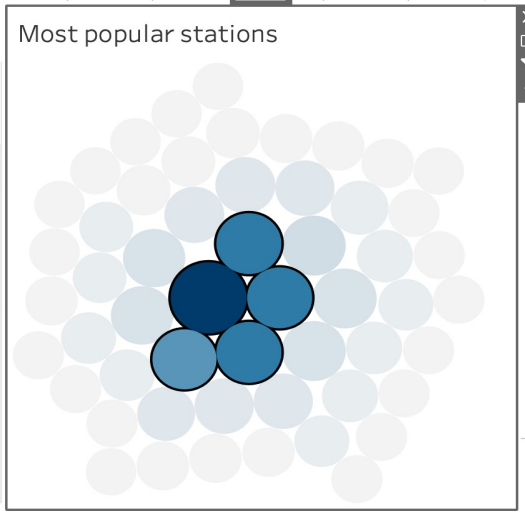
Year
2019
2020

Count of fact_trip
100 226

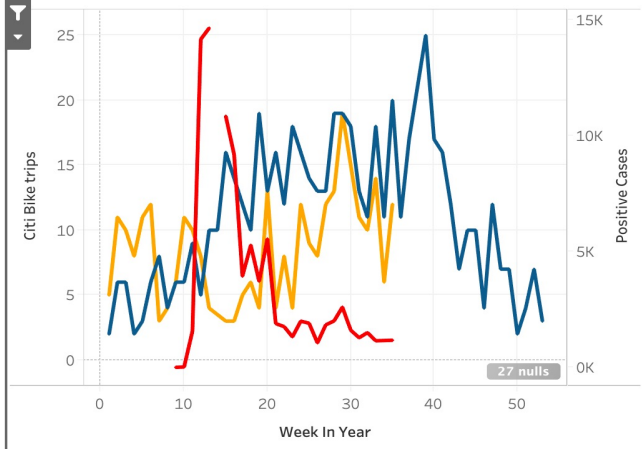
Ridership by station and distance to nearest bus stop



Most popular stations



Trips by Week with Case Count



RECOMMENDATIONS

- Marketing campaign strategy:
  - Customer (non-subscriber)
  - Subscriber
- Marketing campaign highlighting the benefits of Citi bikes over other transportation methods during the pandemic
- Price surging during optimal weather conditions
- Lower prices for bike trips ending in highly demanded stations

Potential Improvements and Future Exploration

- Add more public transportation data
- Explore the frequency of rentals through the day
- Include more data on three fact tables

THANK YOU!

DATA SOURCES

- https://www.citibikenyc.com/system-data
- https://data.cityofnewyork.us/Transportation/Bus-Stop-Shelters/qafz-7myz
- https://www.meteoblue.com/en/weather/archive/export/new-york_united-states-of-america_5128581
- https://www1.nyc.gov/site/doh/covid/covid-19-data.page

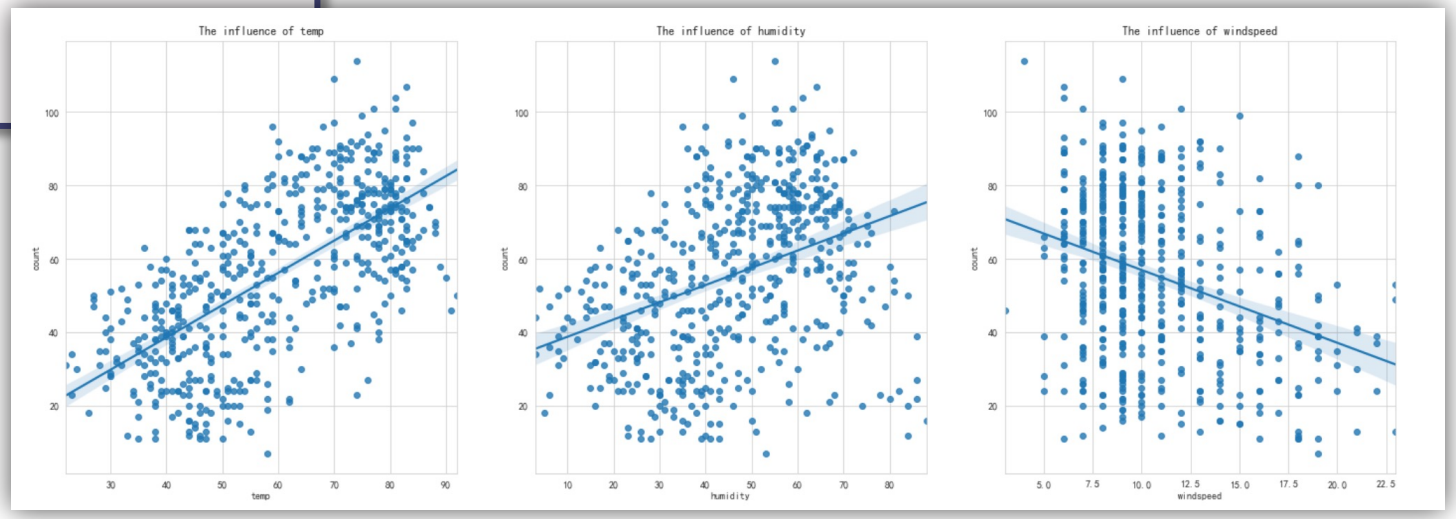
APPENDIX

```sql
select
    min(min_dis) as min_min_dis,
    avg(min_dis) as avg_min_dis,
    max(min_dis) as max_min_dis
from
    nearest_bus_shelter;
```

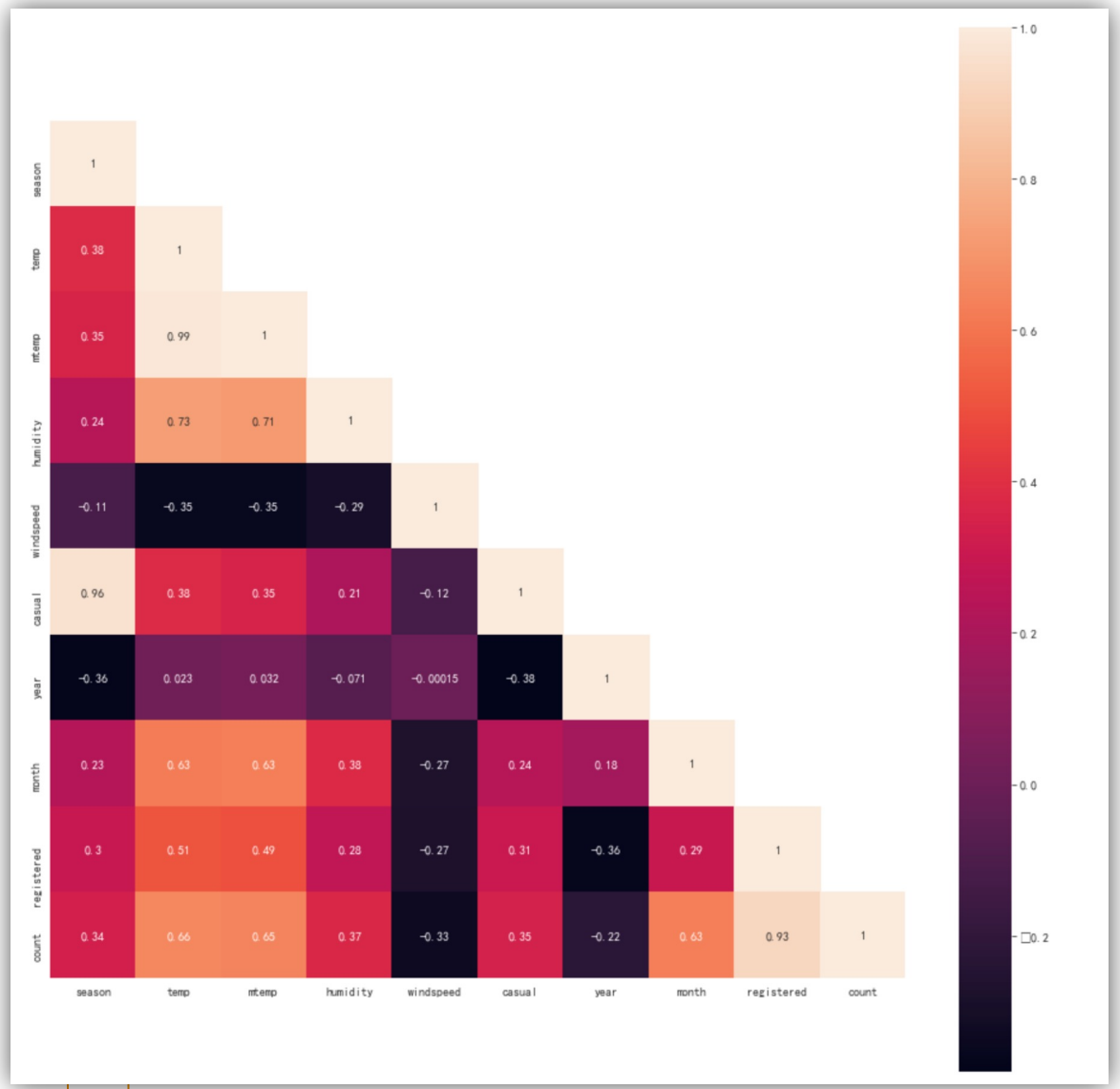
```sql
create view nearest_bus_shelter as
select
    a.has_station_id,
    b.has_shelter_id,
    a.min_dis
from
    (select
        has_station_id,
        min(`distance between`) as min_dis
    from
        dim_bus_stop_station
    group by has_station_id ) a # the nearest bus shelter
        left join
    (select * from dim_bus_stop_station)b on a.min_dis = b.`distance between`;
```

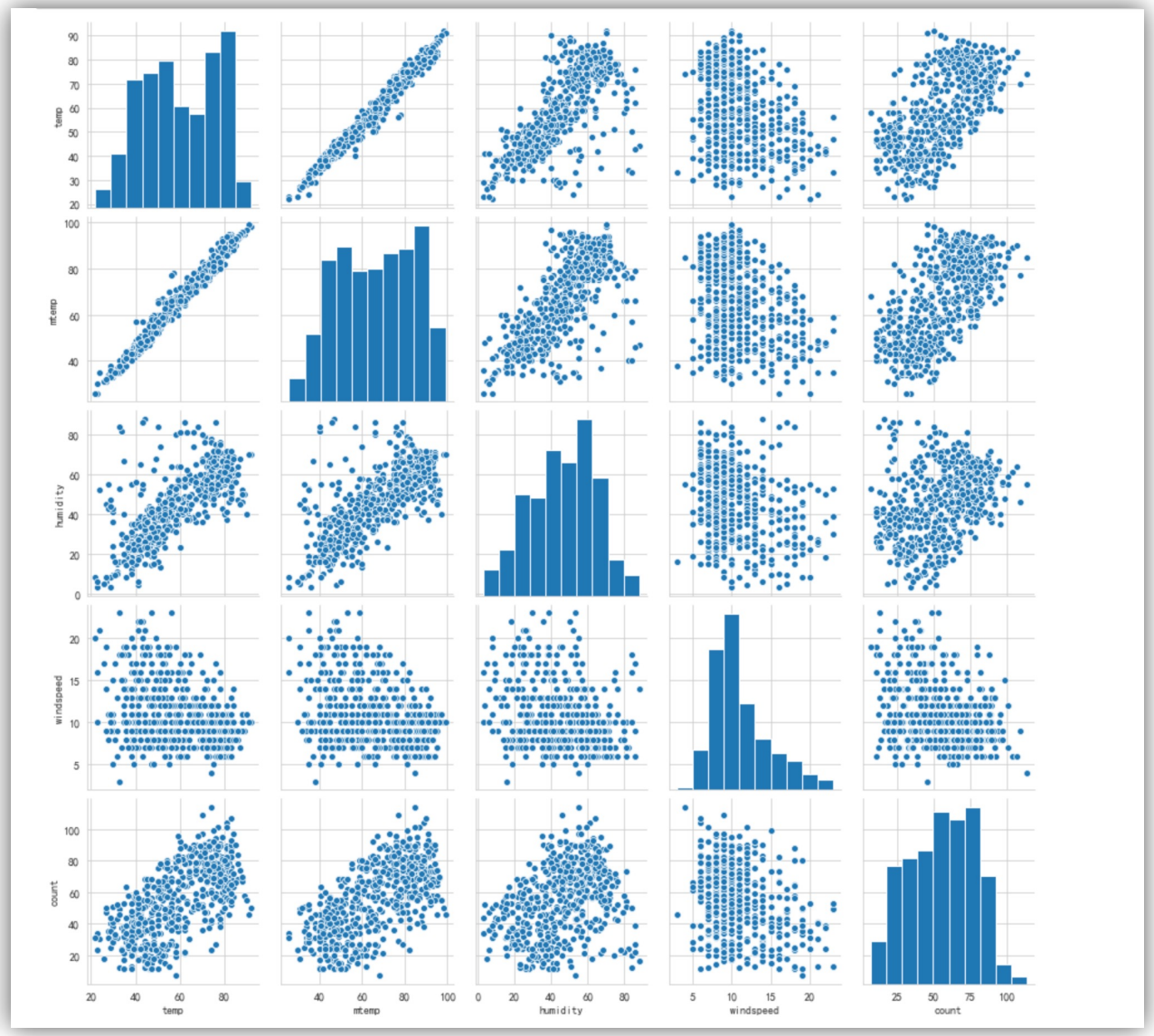
APPENDIX

```sql
select
    weekofyear(fact_covid.date_value) as week_no,
    sum(case_count) case_cnt,
    sum(death_count) death_cnt,
    sum(hospitalized) hospitalized_cnt,
    avg(temp_average) temp_avg,
    avg(humid_average) humid_avg,
    avg(wind_average) wind_avg
from
    fact_covid
    left join
fact_weather on fact_covid.date_value = fact_weather.dim_date_date_value
group by week_no
having week_no > 9
order by week_no;
```



APPENDIX


season
temp
mtemp
humidity
windspeed
casual
year
month
registered
count
1
0.38 1
0.35 0.99 1
0.24 0.73 0.71 1
-0.11 -0.35 -0.35 -0.29 1
0.96 0.38 0.35 0.21 -0.12 1
-0.36 0.023 0.032 -0.071 -0.00015 -0.38 1
0.23 0.63 0.63 0.38 -0.27 0.24 0.18 1
0.3 0.51 0.49 0.28 -0.27 0.31 -0.36 0.29 1
0.34 0.66 0.65 0.37 -0.33 0.35 -0.22 0.63 0.93 1
1.0
0.8
0.6
0.4
0.2
0.0
0.2


temp
mtemp
humidity
windspeed
count